

DESIGN AND CONSTRUCTION OF PROGRAMS PARTICULARLY FOR SPECTRAL ANALYSIS

BERTIL EKENBERG

Lund 1978



DESIGN AND CONSTRUCTION OF PROGRAMS PARTICULARLY FOR SPECTRAL ANALYSIS

BERTIL EKENBERG

FIL. KAND., HB

Akademisk avhandling som för avläggande
av filosofie doktorsexamen vid Matema-
tisk-Naturvetenskaplig fakulteten vid
universitetet i Lund kommer att offent-
ligen försvaras å Institutionen för
Informationsbehandling, Sölvegatan 14 A,
föreläsningssalen, torsdagen den
20 april 1978, klockan 10.15.

Dokumentutgivare
Lund University
Handläggare

Dokumentnamn
Dissertation
Utgivningsdatum

Dokumentbeteckning
LUNFD6/(NFIP-1002)/1-12/
Ärendebeteckning (1978)

Författare
Bertil Ekenberg

Dokumenttitel och undertitel

Design and Construction of Programs Particularly for Spectral Analysis

Referat (sammendrag)

This dissertation describes a computer program, results from the program, and problems on design and implementation of programs presenting graphical results for example from optimizations, especially programs for spectral analysis.

Referat skrivet av

Author

Förslag till ytterligare nyckelord

Curve fitting, models, optimization methods, man-computer interaction, computer graphics, input and output devices.

Klassifikationssystem och -klass(er)

Index termer (ange källa)

Omfattning

12 pages

Språk

English

Sekretessuppgifter

Övriga bibliografiska uppgifter

ISSN

ISBN

Dokumentet kan erhållas från

Lund University
Department of Computer Sciences
Sölvegatan 14 A, S-223 62 LUND/Sweden

Pris

Mottagarens uppgifter

DESIGN AND CONSTRUCTION OF PROGRAMS PARTICULARLY FOR SPECTRAL ANALYSIS

BERTIL EKENBERG

FIL. KAND., HB



Digitized by the Internet Archive
in 2026



This dissertation is based on the following papers:

1. Spektralanalys med hjälp av bildskärm. Report LU/CS 75-3, Department of Computer Sciences, Lund University, Lund, Sweden, 1975. (In Swedish).
2. Results from a Program for Spectral Analysis using a Graphic Display. Report LU/CS 75-5, Department of Computer Sciences, Lund University, Lund, Sweden, 1975.
3. Curve Fitting by Use of Graphic Display, BIT 15 (1975), 385-393.
4. On Design of Programs with the Accent on Programs for Spectral Analysis. Report CODEN:LUNFD6/(NFIP-3003)/1-38/(1978), Department of Computer Sciences, Lund University, Lund, Sweden, 1978.

1. PROBLEM FORMULATION

The problem may be regarded as a curve fitting problem or a parameter estimation problem: Measurements, for example from spectral analysis, may be presented as a curve, as in Fig 1.

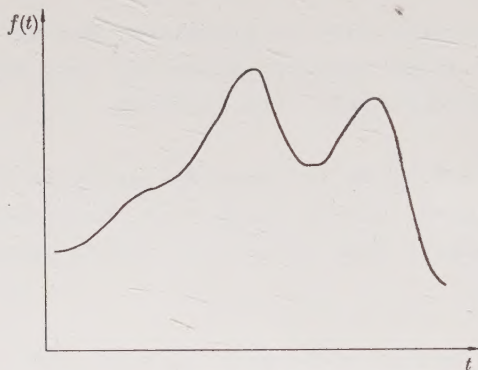


Fig. 1. Example of measurements from spectral analysis, presented as a curve.

If disturbances are neglected, the curve is supposed to depend on certain parameters, for example

$$(1) \quad f(t) = \sum_{k=1}^n \alpha_k \exp(-c(\beta_k - t)^2 / \gamma_k^2)$$

and sometimes a function describing a background, for example a linear function.

The problem is to determine the number of components (n) and the parameters above including the background.

If the function has the form

$$(2) \quad f(t) = \alpha \exp(-c(\beta - t)^2 / \gamma^2)$$

the parameter α corresponds to the height of the peak, β to the position, and γ to the width of the peak. If the constant c is chosen to be $4 \ln 2$, γ is equal to the width at half maximum ($\alpha/2$) of the peak, as illustrated in Fig. 2.

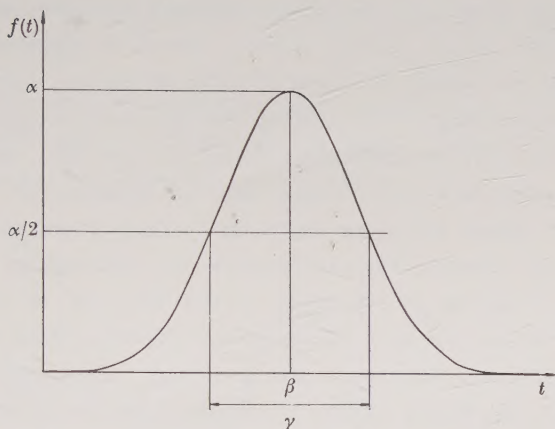


Fig. 2. The meaning of the parameters α , β , and γ when $f(t) = \alpha \exp(-c(\beta - t)^2/\gamma^2)$, $c = 4 \ln 2$.

2. CONSTRUCTION OF A PROGRAM

When the problem defined above was presented to the author, a manual method described in part (1) was used by persons working at the Department of Chemistry at Lund University, but it was believed that better results might be found if a computer was used. Existing programs were run as batch programs with long turn-around time. The author examined programs for spectral analysis by Lederer [2] and [3], but these programs were based on models different from those of the chemists, and for this reason a new program was written. This program is interactive and uses a graphic display system Univac 1557/1558 at Lund Computer Centre.

The graphic display system consists of two units, the display controller and the display console. The display controller may be used as a minicomputer. It has a memory in which picture descriptions and part of the program are stored. The picture is drawn on the display screen as lines and characters. The program in the display unit examines the messages written on the display keyboard, produces the pictures to be displayed, and makes simple calculations.

The display system is connected to the main processor, Univac 1108, at Lund Computing Centre. If more elaborate calculations are required, necessary data are sent to the program in the main processor and the result is sent back to the display unit.

The first version of the program mainly contained routines for reading measurement values, giving starting values or changing old values of the parameters in a sum of Gaussian curves, displaying the corresponding curve, calling the optimization routines Gaussa ([7], [8], and [9]) and Simplex ([5] and [6]), and printing the results on a line printer. When the program was run it sometimes turned out to be difficult to find proper initial values, and hence a routine for that purpose was implemented. Sometimes it was regarded as convenient to fit only part of the curve and this feature was also implemented. Different kinds of data required different function forms. A chemist recommended an optimization routine, Stepit ([1]), which also contains an error estimation routine and provision to fix some parameters to special values. The latter feature was also implemented in the other optimization routines. Still other ideas gradually improved the program, and a detailed description is given in part (1).

3. RESULTS FROM THE PROGRAM

When the program was tested it was convenient to use a constructed curve instead of a measured curve as input. In this way the result of the fit could be compared to the known solution. The form of the curve and the errors could also be controlled, and further the approximate magnitude of some parameters was known beforehand. However, the parameters of the constructed curve should not be known exactly since this knowledge might influence the guess of parameters and choice of method too much.

Therefore a program generating test data was written. The number of components, the parameter values and the perturbations for each function were chosen at random within intervals read from cards. The values defining the curve were stored on a disk memory and then read by the program for spectral analysis, and the actual parameters were written on a line printer.

The test generator and experiments with it are described in detail in parts (1-3). The program usually found correct or almost correct values quickly, but sometimes it was difficult to determine the

parameters. An example of this, also described in parts (2) and (3), is given below.

The curve with the parameters $\alpha_1 = 89.1326$, $\beta_1 = 3.4110$, $\gamma_1 = 1.6671$, $\alpha_2 = 71.7696$, $\beta_2 = 2.8299$, and $\gamma_2 = 1.6119$ was produced. The curve is presented in Fig. 3.

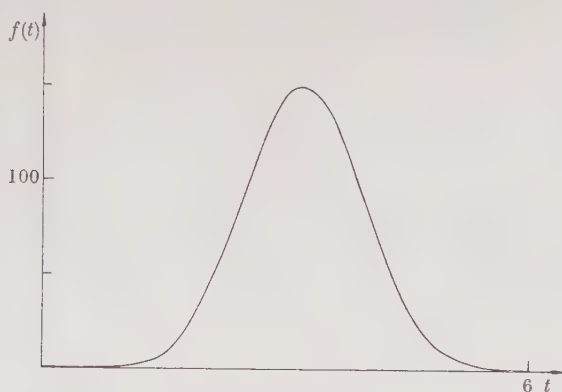


Fig. 3. $f(t) = \sum_{k=1}^2 \alpha_k \exp(-c(\beta_k - t)^2 / \gamma_k^2)$ when $c = 4 \ln 2$, $\alpha_1 = 89.13259$, $\beta_1 = 3.41095$, $\gamma_1 = 1.66706$, $\alpha_2 = 71.76958$, $\beta_2 = 2.82991$, $\gamma_2 = 1.61194$.

Several parameter combinations were found far from the correct values and far from each other, but the differences between these curves and the original curve were very small. Just to mention a few examples, the parameter combinations $\alpha_1 = 94.0153$, $\beta_1 = 3.2125$, $\gamma_1 = 1.8127$, $\alpha_2 = 54.7241$, $\beta_2 = 3.0522$, $\gamma_2 = 1.7241$ and $\alpha_1 = 133.1470$, $\beta_1 = 3.2061$, $\gamma_1 = 1.7766$, $\alpha_2 = 18.0823$, $\beta_2 = 2.7499$, $\gamma_2 = 1.5483$ and $\alpha_1 = 115.8153$, $\beta_1 = 3.2942$, $\gamma_1 = 1.7199$, $\alpha_2 = 41.1807$, $\beta_2 = 2.7373$, $\gamma_2 = 1.5820$ all fit the original curve very well.

The conclusions from this example are that it seems to be impossible to conclude that correct parameter values have been found, even if parameters giving minimum difference between the curves have been obtained, and that it seems to be necessary to know more about the curves to get approximately correct parameter values. Probably constraints describing connections between the parameters should be added to the program. Connections between two or more peaks may be caused by physical or chemical properties of the examined material.

4. ON DESIGN OF PROGRAMS

The conclusions above resulted in further literature studies and discussions with chemists, physicists, numerical analysts, computer scientists, and statisticians on several problems connected with the program design. These and earlier studies and discussions and experience from the construction and use of the program described above resulted in problem formulations and conclusions presented in part (4).

In part (4) the characteristics of a good program is discussed. Although it seems difficult - not to say impossible - to write a program with all good properties one can desire, one ought to give the program as many good properties as possible. Some of the properties of a program for spectral analysis are discussed in the paper.

A part of the problem is the choice of a model. The measured curve is to be fitted to a function, and different function forms may be used depending on the kind of the examined data and the kind of equipment that is used. It is reasonable to choose a model where the parameters correspond to physical or chemical properties of the examined substance and where the meaning of the parameters is clear to the user. Another part of the model formulation problem is to determine whether constraints are needed or not. The difference between the measured curve and the approximating curve may be measured in different ways. The function form used by optimization routines need not be the same as the function presented to the user, so several models may have to be formulated and conversion routines between the models may be needed.

Another part of the problem is how to give starting values of the parameters to be determined by the program, such as α_k, β_k , and γ_k and n in (1). In addition to the possibility to ask the user to supply initial values, there are several numerical methods to locate peaks and to find starting values of the peaks.

When initial values of parameters have been found, one or several optimization routines are usually called. Optimization routines try to find the parameter combination that minimizes the difference (in some sense) between the measured values and an approximating function.

Usually a starting combination of parameters is given, a search direction is computed and a step is taken in that direction which gives a new parameter combination with smaller deviation. This process is repeated until some stop criterion is fulfilled. Several methods to determine step lengths and search directions are mentioned in part (4). Some optimization routines use derivatives which can be computed analytically or numerically. Constraints may be added in different ways to optimization methods, and sometimes it is possible to reformulate the model so that constraints are avoided or more easily handled.

Sometimes one does not only want the parameters corresponding to the optimized function, but also a measure of the reliability of the parameters. Some error estimations are mentioned in part (4).

The routines that make it possible for the user to interact with the program constitute an important part of the running system. A program can be interactive, which means that the user tells the program what to do usually depending on the results from the program. A program can also be a batch program, and in this case the user gives the program all information of what the program should do from the beginning, and when the results are computed the program stops. Whether a program should be interactive or not depends much on whether a co-operation between computer and man gives considerably better results than a computer working alone. If decisions what to do next are often needed and programmable decision rules are difficult or impossible to formulate, the program should be interactive. If the program is interactive, results are more rapidly obtained. But if the peaks of the measured curve in spectral analysis are easy to examine, batch programs may be preferred. The input to a program for spectral analysis is usually the measured values, starting values of parameters, and sometimes other commands telling the program what to do. Different methods to read parameter values and other command are discussed in paper (4). The form and amount of the output is influenced by the resources of devices, the speed of these, the possibility to use the devices interactively, and the capabilities of the operating system of the computer. Suggestions on how to use different devices are given in part (4).

However, there are usually more problems in designing and writing a program for spectral analysis than choosing a model and programming routines for input, output, function evaluations, and optimizations. One such problem is how to represent the parameters that the program is supposed to determine internally in the program. The user may regard the parameters as three one-dimensional arrays or a two-dimensional array, but optimization routines often demand that parameters should be stored in a one-dimensional array. For computational reasons, the area of the peak k may be preferred to α_k above and the parameters should sometimes be scaled to approximately the same magnitude. The user may define a straight line as a line passing through two points, but the optimization routines may prefer the form $f(t) = a t + b$ as a function of a and b . For this reason several representations of parameters and conversion routines between the representations may be needed in the program.

Another problem is whether one or several functions, optimization routines etc should be included in the program. It is sometimes reasonable to permit several function forms, especially if the user changes from one form to another when the program is run. Several optimization routines may be included in the program, for example if one routine is preferred in the case the parameters are far from the correct ones, and another routine if they are almost correct. But using several routines also means more programming, since the program may have to include tests for example to determine which function evaluation routine to use.

More conversion routines or more complicated routines may have to be written. But then the program often becomes more complicated and more difficult to change. Since there are different kinds of spectral analysis with their own problems and models, it is probably preferable to have different programs depending on the kind of spectral analysis and not to design one single program to solve all problems.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Professor Carl-Erik Fröberg for his helpful criticism especially on the presentation of the results, to Doctor Per-Åke Wedin for his continual help with the organization of the work especially his ideas on the design of the program and the choice of numerical methods. Further I want to thank Bengt-Erik Bengtsson for many stimulating discussions on programming problems and related subjects, and Doctor Ragnar Larsson, Roland Lykvist, Eva Andersson och Lars Carlén for valuable discussions on the spectral analysis problem and suggestions for improvements of the program.

REFERENCES

1. J. P. Chandler: Subroutine Stepit. Quantum Chemistry Program Exchange (QCPE), Dept. of Chemistry, Indiana University, Bloomington, Indiana, 1965.
2. C. M. Lederer: Computer Analysis of Spectra. Lawrence Radiation Laboratory, University of California, Berkeley, California. In J. H. Hamilton, J. C. Manthuruthil (ed.): Radioactivity in Nuclear Spectroscopy, Gordon and Breach, New York, 1972, 73-107.
4. C. M. Lederer: Peak-fitting Routines for Spectral Analysis, Lawrence Radiation Laboratory, University of California, Berkeley, California.
5. J. A. Nelder, R. Mead: A Simplex Method for Function Minimization. Computer Journal 7 (1965), 308-313.
6. J. M. Parkinson, D. Hutchinson: An Investigation into the Efficiency of Variants on the Simplex Method. In F. A. Lootsma (ed.): Numerical Methods for Non-linear Optimization, Academic Press, London and New York, 1971, 115-135.
7. P-Å Wedin: The non-linear least squares problem from a numerical point of view. Department of Computer Sciences, Lund University, Lund, Sweden, 1972.
8. P-Å Wedin: On least squares problems with non-linear constraints. Report LU/CS 75-4, Department of Computer Sciences, Lund University, Lund, Sweden, 1975.
9. P-Å Wedin, L. Å. Carlsson: Gaussa. LDC Program Library, Program Abstract, Lund University Computing Centre, Lund, Sweden, 1972 (In Swedish).

